System Verification, Processes and Testing

Report of the Discussions of Breakout Session

Authors: Tristan Perez (Session chair), Andrey Morozov, Børge Rokseth, Jon Arne Glomsrud, Matthew Luckcuck, Thor Myklebust, Tobias Torben, Xue Yang

A challenge related to autonomous systems concern their verification process and testing. This discussion is not detached from regulatory, societal, and ethical requirements. Indeed, being able to verify issues of governance and ethics is of high importance; yet, a key concern is which governance body and whose ethics are being adopted. The verification process should not be entirely removed from these concerns, and ensuring that the right properties are being verified will require interaction with domain experts in those areas. The regulatory, societal, and ethical requirements should be included at the beginning of the design process and should be fed through to the verification phase. However, the verification process may identify ethical concerns (especially if they have not been identified during the requirements and design process) and engineering practice should ensure that these concerns are included into the system's design.

A main concern is how to obtain the right requirements against which to verify the system. While the validation of requirements is a concern with the verification of any system, it may be a particular challenge with autonomous systems. Firstly, this is because of the complexity of autonomous systems; secondly, this is because of a lack of consensus on regulation and ethical guidelines for autonomous systems.

An additional concern is the identification of the best verification processes to use for autonomous systems. Given their complexity, their embodiment in the real world, and their potential for adaptation or learning, continuous and integrated processes are recommended. Briefly, the adoption of a more DevOps-like approach and designing online (continuous or periodic) re-verification systems.

Other elements of the discussion on the topic includes methods for communicating the results of verification efforts and the inclusion of formal methods. Communicating verification efforts to both regulators and the public is important to ensure autonomous systems can be certified, by a regulator, and trusted, by the public. The application of formal methods to the development of autonomous systems can provide automatic verification and unambiguous specification of the system's intended behavior. However, how the autonomy is implemented can have an impact on how challenging the application of formal verification can be.

The following sections deepen those discussions.

Verification Processes for Autonomous Systems

In general, the verification process for autonomous systems should contain firstly an initial verification and testing process and secondly, an on-going process to deal with changes in the system or its operating environment. This can be achieved by adapting classical models such as the V-model into DevOps-like models.

Considering behaviors are the key pathway towards frameworks for certification of autonomous systems. Behavior can be evaluated in terms of safety, performance and ethics. The process of initial verification and testing consists of first identifying desired behaviors for safety, performance, security and ethics. Then metrics and verification criteria must be established before the actual verification and testing activities take place. After being built, verified, and accepted a system may change due to, for example, software updates or any potential learning ability of the system. In addition, the system environment may change. For example, an autonomous car may be taken to a new area where other cars and pedestrians behave differently. The on-going verification process is intended to deal with such changes. In order to achieve this, changes must be detected and analyzed to determine the effect in terms of verification needs. The verification and testing process can be discussed in terms of three steps:

1. Defining desired behavior for the autonomous system;
2. Identifying and conducting tests and verification to satisfy verification criteria;
3. Monitoring systems and conducting change analysis during operation to detect any new needs for verification due to system or environmental changes.

Step 1

When defining the desired behavior for the autonomous system, it is necessary to determine a level of granularity at which the desired situational behaviors are defined. This raises questions, such as, to which levels systems should be decomposed and how systems should be decomposed. In general, desired behavior should be specified at the system level and then refined as much as necessary into components or sub-functions in order to determine sub-system or sub-function behavior that ensures the desired system level behavior.

Methods for specifying desired behavior may need to be specified case by case. It is reasonable to assume, however, that elements of hazard analysis (i.e. identifying what can go wrong and how the system should handle these situations) as well as formalizing design requirements and requirements from standards into behavioral models will be highly relevant approaches. An important part of the documentation of this step will be to record the assumptions made regarding the system and its operating environment.

Step 2

The next step of the verification process for autonomous systems, is to identify and conduct tests and verification to satisfy verification criteria. The goal of this step is to observe the system behavior under tests and other verification activities, and to evaluate the observed behavior to determine our confidence that the system will behave according to the desired behavior. Increasing this confidence corresponds to reducing uncertainty. There are two general types of uncertainty in this context. First, there is uncertainty about whether an observed behavior should be classified as desired or undesired behavior, and secondly, there is uncertainty when a certain behavior is observed in one scenario related to the extent to which this can be considered representative for similar scenarios. Verification then is about collecting evidence to reduce these uncertainties. To achieve this, verification needs both to be broad in terms of capturing as many types of scenarios as possible while it also is necessary to test each type of scenario extensively to ensure that results are representative of all similar scenarios. A formal verification, model-in-the-loop, process-in-the-loop and hardware-in-the-loop methods may be central methods for collecting evidence to reduce uncertainty and increase confidence.

Verification of autonomous systems may be more resource demanding than verification of traditional systems because there will be more focus on system behavior and there can be a huge number of possible behaviors. It will be more critical for autonomous systems than for human operated systems to foresee abnormal scenarios because the autonomous systems may be less robust and innovative with respect to handling the unforeseen. Therefore, any possible scenarios must be foreseen and considered in the verification process. This may cause state explosions and the necessity for rare event simulations.

Step 3

The third step of verification is to monitor operations and detect emerging verification needs during the operational phase of the system. One important aspect of this is to define the operational environment for which the system has been verified, as well as the system that has been verified. The assumptions being made regarding the system and its operational environment must hold true for each conclusion reached during verification to be a valid verification. Once these assumptions are known they can be monitored during the operational phase of the system and if one no longer holds true, further verification is necessary. Change analysis is proposed to achieve this.

Communicating Verification Results

The group discussed the challenge of communicating the results of verification to stakeholders –regulators and the public.

Some sectors in which autonomous systems are being explored require that a system is certified. Regulators need to be able to understand how an autonomous system is verified (and be confident in the verification results) in order to certify a system for use. Given the complexity of autonomous systems and their potential to change (either through learning, self-reconfiguration, or simply by changing their operational environment) efforts must be made to ensure that verification approaches for autonomous systems are amenable to the regulator(s) of the sector in which they are to be deployed.

Communicating the concept and results of verification to public is key to gaining public trust of autonomous systems. Society seems to have lower tolerance for accidents and unexpected behavior from autonomous systems, so efforts to ensure public trust should help with the adoption of autonomous systems in meaningful use cases within society. Results from verification must be interpreted and presented in a way that helps decision-making, such as, whether it is safe to deploy a system into society. Other key challenges here are how to communicate the level of confidence and uncertainty in the system's ability to continue to operate according to desired behavior, and how to communicate what the desired in a digestible way.

Formal Methods

Formal methods are mathematically defined techniques to the specification, design, and verification of computer systems and software. They enable the expression of requirements and description of systems with precision and no ambiguity. Often the tool support for checking that a system exhibits the required properties is automatic and exhaustive. The formal specification and verification of autonomous robotic systems is an ongoing topic of research for the formal methods community.

The successful application of formal methods to autonomous systems can largely depend on how the autonomy is implemented. Neural networks, for example, are challenging for formal methods to deal with because it is often not understood how they produce their output. Formal methods work best with more symbolic approaches to autonomy.

Arguments were put forth arose that including formal methods into the specification, design, and verification of autonomous systems is very important because of their increasingly safety-critical nature. Formal Methods can be introduced at several stages during the development process. For specification, they can help to clarify the requirements (and even check that the requirements themselves have not introduced unintended errors). During design, they can be used to check that the designs meet the requirements. During verification, various automatic tools exist to exhaustively check that the description of the system preserves the required (safety, legal, ethical, etc.) properties. This automation will help with the DevOps-like process of ongoing verification described above.

An obvious final challenge is that of ensuring that the final system represents the formal descriptions of the system, and so preserves the required properties. This is a challenge faced by any software development process. Some formal methods can verify program code (for example the Agent Java Pathfinder, a program model checker for agent-based autonomous systems) and there are other methods from which program code can be automatically generated. Even without these types of method, using formal methods during the requirements and design phases can help to reduce errors introduced at these early stages of the development process.

Conclusion

Six main challenges and four distinct opportunities related to verification and testing of autonomous systems can be pointed out. The following challenges were identified:

1. The V-model may no longer be adequate and is necessary to either replace it or adapt it into a DevOps-like model.
2. Autonomous systems may sometimes need assistance from operators, and in certain scenarios, control needs to be handed over from the autonomous system to the operator. Verification of the control handover may be a particular challenge
3. In traditional systems, the behavior of the system is to a greater extent governed by human operators than what will be the case for autonomous systems. Operators are often trained and certified, and together with their general human experience. This is accepted as sufficient. Once the system behavior starts being governed by software, rather than human operators, how does this process

translate to training and certification of human operators and the consequent level of trust?

4. Learning algorithms may be central to autonomous systems control. A specific verification challenge is how can trust in a system be established that may continue to adapt itself after deployment. Thinking of the verification process as ongoing through the life cycle of a system will be a central issue with respect to this challenge.
5. It will also be a challenge to formulate and parametrize desired behaviors. It may be close to impossible to cover all operational profiles. While systems operated by humans have a certain robustness because they can adapt to situations, and as such can handle unforeseen scenarios, autonomous systems are not robust in this sense. This means that any scenario must be foreseen, and a system response must have been planned for the system to be able to handle this situation.
6. In order to cope with a huge number of scenarios, automated and customizable methods and tools for verification and testing must be developed.

While there are challenges related to verification of software rather than human operators, who are governing the behavior of systems, there are also opportunities related to this. In addition to the six challenges, four main verification and testing opportunities for autonomous systems are identified:

1. When the human operator is replaced by software, this enables replacing periodic inspections with continuous performance monitoring which can be used to revoke operating license in the event of inadequate performance.
2. The behavior of software can be considered more deterministic compared to human operators. In general, it is believed that it is possible to predict the behavior of software with higher precision than that of human operators. While it is not possible to inspect an operator’s brain to determine how the operator will respond to different inputs, it is possible to inspect the software code to determine this.
3. Once the human operator is out of the loop, it is possible to predict and verify behavior online through online model-based verification where variations of the current operational scenario can be simulated into the future to verify safe system response.

4. With human operators in the loop, automated accelerated testing of the complete system is not possible. With the human out of the loop, testing can be conducted in simulators faster than real-time.

Group Participants

Andrey Morozov
Technical University, Dresden, Germany

Børge Rokseth
Department of Marine Technology, NTNU, Norway

Jon Arne Glomsrud
DNV GL, Norway

Matthew Luckcuck
University of Liverpool, United Kingdom

Thor Myklebust
SINTEF Digital, Norway

Tobias Torben
Department of Marine Technology, NTNU, Norway

Tristan Perez
Boeing Research and Technology, Australia

Xue Yang
Department of Marine Technology, NTNU, Norway